# An Algorithm for Merging Hyperellipsoidal Clusters

Patrick M. Kelly
Los Alamos National Laboratory

October 1, 1994

**Abstract**

This report discusses an algorithm for merging hyperellipsoidal clusters. The *effective merging radius* between two clusters is introduced, and this measure is used to determine the order in which clusters are combined. We continue to merge clusters until a desired number of clusters is reached, or until the minimum *effective merging radius* is larger than some predetermined threshold.

## 1 Introduction

Clustering algorithms are used for a variety of applications including pattern recognition, data compression, and data analysis. Although it can be straighforward to find representative clusters for a given data set when the number of natural clusters in the data is known *a priori*, it is difficult to automatically determine the number of clusters when it is not given. It is therefore useful to use a large number of clusters for the initial clustering of the data, after which a cluster merging algorithm can be employed to find a smaller set of representative clusters.

The k-means clustering algorithm [1] uses a Euclidean distance measure to determine the distance between two data samples, and therefore attempts to group data into hyperspherical regions. The positions for each cluster are automatically generated by the k-means algorithm. Once k-means has finished, we calculate a covariance matrix for each cluster, giving us a hyperellipsoidal representation for each cluster, and therefore a starting point for the merging algorithm which we will present. Initial clusters containing only a few outlying data samples or having a singular covariance matrix are deleted from the system and ignored in subsequent processing.

In this report, we will assume that we are given a set of $K$ hyperellipsoidal clusters as a starting point. The position of each cluster is specified by a mean vector, $\underline{\mu}_i$. The shape, size, and orientation of each cluster is specified by a covariance matrix $\Sigma_i$. The relative size of each cluster with respect to the other clusters in the set is determined by the number of elements assigned to that cluster, $N_i$. Our goal is to reduce the number of clusters needed to adequately describe the distribution of the data that was clustered. This merging algorithm was first described in [2] and [3].

## 2 Cluster Merging

Our cluster merging algorithm tries to identify a pair of clusters, from our initial set of $K$ clusters, that can be combined into a single cluster. The single cluster that results from merging two separate clusters needs to adequately

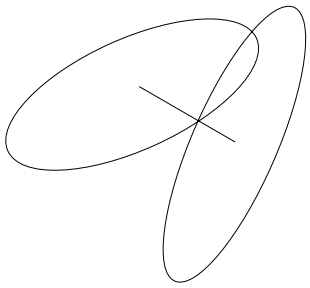

Figure 1: Defining Effective Merging Radius for Clusters $i$ and $j$

represent the distribution of the underlying data. Once this pair of clusters is combined into a single cluster, the algorithm continues by selecting the next pair of clusters that are likely candidates to be merged.

When we combine two clusters into a single cluster, it is straightforward to determine the parameters of the resultant cluster without recalculating statistics from the original data. If clusters are defined by a mean vector, $\underline{\mu}_i$, covariance matrix, $\Sigma_i$, and the number of cluster elements, $N_i$, then combining clusters $i$ and $j$ produces a new cluster with the following statistics:

$$N_{new} = N_i + N_j \tag{1}$$

$$\underline{\mu}_{new} = \frac{N_i}{N_{new}}\underline{\mu}_i + \frac{N_j}{N_{new}}\underline{\mu}_j \tag{2}$$

$$\Sigma_{new} = \frac{N_i - 1}{N_{new} - 1}\Sigma_i + \frac{N_j - 1}{N_{new} - 1}\Sigma_j + \frac{N_i N_j}{N_{new}(N_{new} - 1)}[(\underline{\mu}_i - \underline{\mu}_j)(\underline{\mu}_i - \underline{\mu}_j)^T] \tag{3}$$

# 3 Effective Merging Radii

There are numerous algorithms for merging clusters that one might consider [4, 5]. Many of these rely on a Euclidean distance metric as a measure of similarity between clusters. Unfortunately, these methods do not necessarily take into account the actual spread of the data around each cluster centroid.

Before we can develop a merging algorithm, we need to answer the following question: "Given a set of $K$ clusters, which two clusters can be most accurately represented by a single cluster?" To answer this question, we will treat each cluster as a Gaussian probability density function. This is possible since each cluster is already specified by a position ($\underline{\mu}_i$) and a shape ($\Sigma_i$). We can then suggest that the two clusters most likely to be merged should be "close" to one another, and there should be a reasonable amount of data lying between the two; there should not be a significant "gap" where no data samples exist.

To answer the question we just posed, we will introduce the concept of an *effective merging radius*. We will use $r_{ij}$ to denote the *effective merging radius* between clusters $i$ and $j$. We define $r_{ij}$ to be the minimum effective cluster radius such that the boundary of hyperellipsoids $i$ and $j$ intersect on the segment between their mean vectors (see

Figure 1). After considering all possible pairs of clusters in our data set, the two clusters yielding the smallest $r_{ij}$ will be merged. We can find this value as follows. First, we know that this is an effective cluster radius for both hyperellipsoids, so a common point $\underline{x}$ lying on both boundaries must satisfy:

$$(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1} (\underline{x} - \underline{\mu}_i) = r_{ij} \tag{4}$$

$$(\underline{x} - \underline{\mu}_j)^T \Sigma_j^{-1} (\underline{x} - \underline{\mu}_j) = r_{ij} \tag{5}$$

Since we are looking for an $\underline{x}$ which lies between $\underline{\mu}_i$ and $\underline{\mu}_j$, we can express it as:

$$\underline{x} = \alpha \underline{\mu}_i + (1 - \alpha) \underline{\mu}_j \tag{6}$$

In this equation, $0 < \alpha < 1$. Substituting relationship (6) into (4) and (5) and simplifying, we get:

$$\alpha = (-k_2 \pm \sqrt{k_1 k_2})/(k_1 - k_2) \tag{7}$$

$$k_1 = [\underline{\mu}_i{}^T \Sigma_j^{-1} \underline{\mu}_i + \underline{\mu}_j{}^T \Sigma_j^{-1} \underline{\mu}_j - 2\underline{\mu}_i{}^T \Sigma_j^{-1} \underline{\mu}_j] \tag{8}$$

$$k_2 = [\underline{\mu}_i{}^T \Sigma_i^{-1} \underline{\mu}_i + \underline{\mu}_j{}^T \Sigma_i^{-1} \underline{\mu}_j - 2\underline{\mu}_i{}^T \Sigma_i^{-1} \underline{\mu}_j] \tag{9}$$

Now since $0 < \alpha < 1$ and $k_1$ and $k_2$ are both positive, we get:

$$\alpha = \frac{\sqrt{k_1 k_2} - k_2}{k_1 - k_2} \tag{10}$$

In summary, one must calculate the values $k_1$ and $k_2$ from the cluster statistics, use these values to find $\alpha$, and then calculate $r_{ij}$:

$$r_{ij} = k_1 \alpha^2 \tag{11}$$

# 4 Termination

So, given this method of calculating $r_{ij}$, we have specified an order for merging clusters. Depending on the application, the merging process is halted by either of two methods. One method is to reduce the current set of clusters to a given size. Another method would be to specify a threshold value $R$, and use it to determine when merging should stop. In this case, merging will only occur as long as the minimum value for $r_{ij}$ is smaller than $R$. As $R$ increases, the final number of clusters will decrease.

Using this second method allows a user to pick a value for $R$ in such a way that an optimal number of clusters is produced. The way we propose doing this is to treat each cluster as a Gaussian distribution function. In this case, $R$ can be selected in such a way that a confidence interval containing some percentage of the data falls outside of this region.

Of course, many cluster merging algorithms have been developed in the past, and this order to merging clusters may be incorporated into them. For instance, the clustering algorithm proposed in [6] attempts to merge any two randomly selected clusters from the same class at any given iteration. Using our method of calculating $r_{ij}$ could give a more meaningful order to this algorithm.

# 5 Example

Figure 2 displays a set of sample data that was generated by four separate Gaussian random number generators. This data has been clustered to 20 clusters via the k-means clustering algorithm. The left-hand side of the figure shows the clusters before any merging takes place. After merging this set down to 4 clusters, we have a better representation of the data, and in fact we have identified the four separate Gaussian modes used to generate the data.

# 6 Conclusions and Future Work

This method for merging clusters looks not only at cluster positions, but also at cluster variances to determine where clusters should be merged and where they should be left separated. The *effective merging radius* provides a mechanism that allows us to determine the order that merging should occur for a given set of hyperellipsoidal clusters.

One of the most important aspects of this problem that needs to be explored is the method for selecting a threshold value $R$ to terminate the merging process. Given a robust method for choosing $R$, this algorithm promises to serve as a powerful tool for determining a good, representative set of hyperellipsoidal clusters for a give data set.

# References


[1] J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley, Reading, MA, 1974.

[2] P.M. Kelly. *A One-Class Classifier Using Hyperellipsoidal Decision Surfaces.* Masters Thesis, University of New Mexico, 1991.

[3] P.M. Kelly, D.R. Hush, and J.M. White. An adaptive algorithm for modifying hyperellipsoidal decision surfaces. *Journal of Artificial Neural Networks.* In Press.

[4] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, 1973.

[5] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[6] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels. On the training of radial basis function classifiers. *Neural Networks*, 5:595–603, 1992.

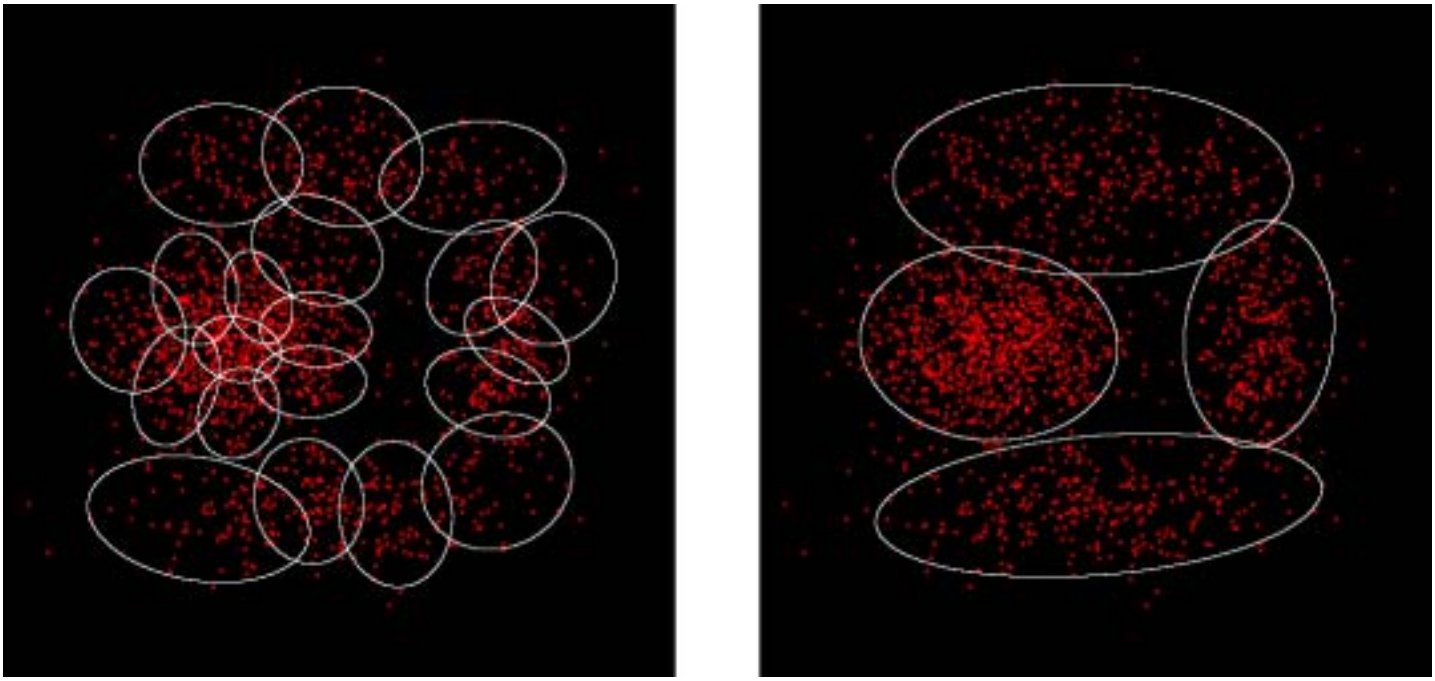

Figure 2: Sample data. All cluster boundaries shown contain 90% of the data for a Gaussian distribution. The data set was originally clustered to 20 clusters (left). After merging to a set of 4 clusters, a much more reasonable representation is obtained. The resulting clusters describe the underlying distribution of data samples well.

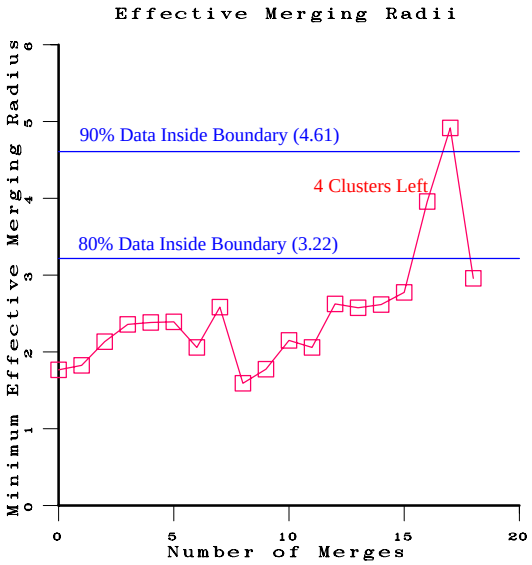


Figure 3: Plot of effective merging radii. This plot shows the minimum effective merging radius as cluster consolidation takes place. As we reach a "good" solution, these values become large, indicating that empty areas in the data space exist between clusters.